Claims **1-29** are pending.

1. **(Currently amended)** A method for managing topology changes in media applications, comprising:

receiving a partial media topology that <u>defines how data flows through</u> ~~includes~~ a plurality of nodes including at least a first media source node and at least a first media sink node;

retrieving a cached media topology that <u>defines how data flows through</u> ~~includes~~ a plurality of nodes including at least a second media source node, at least a second media sink node, and at least one transform node; and

copying one or more nodes from the cached media topology to the partial media topology.

2. **(Original)** The method of claim 1, wherein the partial media topology is received from a remote process as a parameter in an interface call.

3. **(Original)** The method of claim 1, wherein the cached media topology is retrieved as a parameter in an interface call.

4.   **(Currently Amended)** The method of claim 1, further comprising determining whether there are corresponding nodes in the partial <u>media </u>topology and the cached topology.

5.   **(Currently Amended)**   The method of claim 4, further comprising transferring the at least one transform node from the cached <u>media </u>topology to the partial <u>media</u> topology.

6.   **(Currently Amended)**   The method of claim 1, further comprising cloning a plurality of connected nodes from the cached <u>media </u>topology into the partial <u>media</u> topology.

7.   **(Currently Amended)** The method of claim 1, further comprising maintaining a data table that correlates one or more decoders in the cached <u>media </u>topology with one or more source nodes in the cached <u>media </u>topology.

8.   **(Currently Amended)**   The method of claim 1, further comprising connecting one or more nodes in the partial <u>media </u>topology.

9.   **(Currently Amended)**   The method of claim 8<u>,</u> wherein connecting the one or more intermediate nodes between the <u>first </u>media source node and the <u>first </u>media sink node comprises generating a data path between <u>an </u>~~the~~ output of a node <u>and </u>an input of an intermediate node.

10.   **(Currently amended)** A system comprising:

one or more computer-readable storage media; and

a media engine embodied on the one or more computer-readable media and configured to communicatively interact with an application to present a media presentation;

the media engine being configured to use:

a media session to generate a partial topology, the partial topology including one or more media sources, individual ones of which serve as a source of media content, and one or more media sinks configured to sink a media stream;[[,]] and

a topology loader to resolve the partial topology into a full media topology, wherein the topology loader is configured to copy one or more nodes from a cached media topology to resolve the a resolved full media topology, and the topologies define a flow of data through the nodes.

11.   **(Original)**   The system of claim 10, wherein the media session passes the partial topology to the topology loader as a parameter in an interface call.

12.   **(Original)**   The system of claim 10, wherein the media session passes the cached topology to the topology loader as a parameter in an interface call.

13.   **(Original)**   The system of claim 10, wherein the topology loader is configured to determine whether there are corresponding nodes in the partial topology and the cached topology.

14.   **(Original)**   The system of claim 10, wherein the topology loader is configured to clone one or more intermediate nodes from the cached media topology, and to connect the one or more intermediate nodes in a communication path between a media source and a media sink in a partial topology.

15.   **(Currently Amended)**   The system of claim 14, wherein the one or more intermediate nodes comprise a decoder for decoding <u>an</u> ~~the~~ output of a source node.

16.   **(Original)**   The system of claim 14, wherein the one or more intermediate nodes comprises an encoder for encoding an input of a source node.

17.   **(Original)**   The system of claim 10, wherein the topology loader is configured to maintain a data table that associates one or more decoder nodes with a source node from one or more previous topologies.

18.   **(Original)**   The system of claim 10, wherein the topology loader maintains a data table that stores one or more encoder nodes from one or more previous topologies.

19.   **(Original)**   The system of claim 10, wherein the topology loader returns a fully resolved topology to the media session.

20.   **(Currently amended)** One or more computer-readable <u>storage</u> media comprising computer executable instructions that, when executed on a computer, direct the computer to:

      receive a partial media topology <u>defined by the flow of data through various components</u> that includes a plurality of nodes including at least a first media source node and at least a first media sink node;

      retrieve a cached media topology that includes a plurality of nodes including at least a second media source node, at least a second media sink node, and at least one transform node; and

      copy one or more nodes from the cached media topology to <u>a</u> ~~the~~ full<u>y</u> resolved media topology.

21.   **(Currently Amended)**   The one or more computer-[[ ]]readable <u>storage</u> media of claim 20, wherein the partial media topology is received from a remote process as a parameter in an interface call.

22.   **(Currently Amended)**   The one or more computer-[[ ]]readable <u>storage</u> media of claim 20, wherein the cached media topology is retrieved as a parameter in an interface call.

23.   **(Currently Amended)**   The one or more computer-[[ ]]readable <u>storage</u> media of claim 20, further comprising computer executable instructions that, when executed on a computer, direct the computer to determine whether there are corresponding nodes in the partial topology and the cached topology.

24.   **(Currently Amended)**   The one or more computer-[[ ]]readable <u>storage</u> media of claim 20, further comprising computer executable instructions that, when executed on a computer, direct the computer to transfer the at least one transform node from the cached topology to the partial topology.

25.   **(Currently Amended)**   The one or more computer-[[ ]]readable <u>storage</u> media of claim 20, further comprising computer executable instructions that, when executed on a computer, direct the computer to clone a plurality of connected nodes from the cached topology into the partial topology.

26.   **(Currently Amended)**   The one or more computer-[[ ]]readable <u>storage</u> media of claim 20, further comprising computer executable instructions that, when executed on a computer, direct the computer to maintain a data table that correlates one or more decoders in the cached topology with one or more source nodes in the cached topology.

27.   **(Currently Amended)** The one or more computer-[[ ]]readable storage media of claim 20, further comprising computer executable instructions that, when executed on a computer, direct the computer to connect one or more nodes in the partial topology.

28.   **(Currently Amended)** The one or more computer-[[ ]]readable storage media of claim 20, further comprising computer executable instructions that, when executed on a computer, direct the computer to generate a data path between an the output of an upstream node and an input of a downstream node.

29.   **(Currently amended)**   A topology loader module comprising computer executable instructions stored in computer-readable storage media that, when executed by a computer, provide:

        means for receiving a partial media topology that defines how data flows through includes a plurality of nodes including at least a first media source node and at least a first media sink node;

        means for retrieving a cached media topology that defines how data flows through includes a plurality of nodes including at least a second media source node, at least a second media sink node, and at least one transform node; and

        means for copying one or more nodes from the cached media topology to a the fully resolved media topology.